Requêtes Skyline Hiérarchiques

Tassadit Bouadi*, Marie-Odile Cordier*, René Quiniou**

*IRISA - Université de Rennes 1, Campus de Beaulieu, 35042 RENNES, France
tassadit.bouadi@irisa.fr, marie-odile.cordier@irisa.fr
**IRISA - INRIA Rennes, Campus de Beaulieu, 35042 RENNES, France
rene.quiniou@inria.fr

Résumé. Les requêtes skyline constituent un puissant outil d'analyse de données multidimensionnelles et d'aide à la décision. Lorsque les dimensions sont conflictuelles, les requêtes skyline retournent les meilleurs compromis associés à ces dimensions. De nombreux travaux se sont intéressés à l'extraction de points skyline dans le contexte de bases de données multidimensionnelles, mais, à notre connaissance, aucun de ces derniers n'a traité la problématique des skyline associés à des données agrégées lorsque les dimensions sont multiples et hiérarchiques. Cet article présente une méthode qui étend la recherche des meilleurs compromis aux dimensions hiérarchiques. Notre proposition *HSky* (Hierarchical Skyline Queries) se focalise sur la définition de la sémantique et du calcul efficace de points skyline sur des dimensions présentant plusieurs niveaux hiérarchiques.

1 Introduction

Les requêtes skyline constituent un puissant outil d'analyse de données multidimensionnelles et d'aide à la décision. De nombreux travaux tels Raïssi et al. (2010); Wong et al. (2009); Tao et al. (2008); Jin et al. (2007b); Huang et al. (2008); Borzsonyi et al. (2001) se sont intéressés à l'extraction de points skyline dans le contexte de bases de données multidimensionnelles, mais aucun de ces derniers n'a traité la problématique des skyline associés à des données agrégées lorsque les dimensions sont multiples et hiérarchiques comme dans les entrepôts de données. Nous souhaitons coupler l'analyse skyline à l'analyse en ligne afin de permettre à l'utilisateur de naviguer vers les faits les plus intéressants de l'entrepôt de données. Dans cet objectif, il faut calculer les skyline en présence de dimensions hiérarchiques. Pour résoudre ce problème, plusieurs verrous scientifiques et techniques sont à lever. Peut-on les dériver à partir des skyline de niveau supérieur ou inférieur ? Peut-on étendre les algorithmes existants de calculs de skyline en tirant parti des dimensions hiérarchiques ? Doit-on recalculer tous les skyline pour chaque niveau hiérarchique ?
Dans la littérature, quelques récents travaux comme Antony et al. (2009, 2008); Jin et al. (2007a) se sont intéressés à l'application des requêtes skyline sur des données agrégées. Ces travaux sont principalement axés sur l'optimisation de requêtes sollicitant les deux opérateurs *Skyline* et *Group-By*, ce dernier étant un opérateur d'agrégation dans les bases de données. Cependant, ces approches se sont contentées d'exécuter séquentiellement ces deux opérateurs

sans réel couplage. Calculer le skyline avant d'appliquer l'opérateur *Group-By* revient à appliquer le skyline à des points individuels sans prendre en compte les connaissances liées à la structure de leur groupe. De même, un skyline appliqué avant un *Group-By* se réduit à l'accès au résumé des informations calculées par les fonctions d'agrégation.

De manière plus intéressante, l'opérateur *Aggregate Skyline* proposé par les auteurs de Magnani et Assent (2013), combine les fonctionnalités des deux opérateurs *Skyline* et *Group-By*. La relation de dominance est adaptée au concept de groupe afin de parvenir à extraire l'ensemble des groupes non dominés par d'autres groupes. En d'autres termes, cet opérateur permet à l'utilisateur de poser des requêtes skyline sur des groupes de points afin de sélectionner le groupe le plus pertinent.

Dans cet article, nous étudions le cas des préférences associées à des dimensions hiérarchiques dans le contexte des skyline. À partir d'une préférence de base et de l'ensemble skyline associé, nous souhaitons permettre à l'utilisateur de spécialiser ou de généraliser cette préférence et de dériver le skyline correspondant tout en respectant la structure hiérarchique des dimensions concernées.

Une solution naïve serait, soit de recalculer à chaque niveau hiérarchique l'ensemble des skyline, soit de stocker pour chaque niveau hiérarchique l'ensemble des skyline associés. Cependant, ces solutions sont très coûteuses en temps de calcul et/ou de stockage mémoire dans un contexte de données volumineuses. Nous proposons donc une méthode $HSky$ *(Hierarchical Skyline queries)* permettant de réduire l'espace mémoire requis tout en assurant un calcul efficace et interactif des points skyline à tous les niveaux hiérarchiques des différentes dimensions. Nous mettons en évidence un ensemble de propriétés permettant la formalisation des relations hiérarchiques entre les préférences associées aux différentes dimensions. Ces propriétés permettent notamment aux utilisateurs de naviguer en spécialisant ou en généralisant leurs préférences et d'extraire les points skyline associés de manière efficace.

Dans cet article, nous présentons dans la section 2 les concepts de base relatifs aux requêtes skyline ainsi qu'aux propriétés liées aux dimensions hiérarchiques et aux préférences associées à ces dernières. Dans la section 3, nous détaillons les aspects formels ainsi que l'implémentation de notre proposition $HSky$ qui étend la recherche des points skyline aux dimensions hiérarchiques. Dans la section 4, nous présentons les résultats de l'évaluation expérimentale réalisée sur des données synthétiques qui souligne la pertinence de la solution proposée.

2 Concepts de base

Soit $D = \{d_1, ..., d_n\}$ un espace à n dimensions, E un ensemble de points définis dans l'espace D et p, q deux points de l'ensemble E. On note par $p(d_i)$ la valeur de p sur la dimension d_i. Une préférence sur le domaine de valeurs de la dimension d_i, notée par $\wp_{d_i}$, est définie par un ordre partiel $\leqslant_{d_i}$. Cet ordre peut aussi se représenter par l'ensemble des préférences binaires (ensemble éventuellement infini) $\wp_{d_i} = \{(u, v) | u \leqslant_{d_i} v\}$, avec (u, v) un couple de valeurs. $\wp = \bigcup_{i=1}^{|D|} \wp_{d_i}$ désigne l'ensemble des préférences associées à l'espace D. Dans la suite de cet article, nous ne considérons que les fermetures transitives des préférences. Par exemple, à la préférence $\{(Yar, Epte), (Epte, Yeres)\}$ nous ferons correspondre sa fermeture transitive : $\{(Yar, Epte), (Epte, Yeres), (Yar, Yeres)\}$.

Dans l'espace D, un point p $domine$ un autre point q, noté par $p \prec_D q$, si $\forall d_i \in D$, $p(d_i) \leqslant_{d_i} q(d_i)$ (i.e. p est préféré ou égal à q dans D) et $\exists d_i \in D, p(d_i) <_{d_i} q(d_i)$ (i.e. p est strictement préféré à q sur d_i). Pour plus de lisibilité, $p \prec_D q$ sera simplement noté $p \prec q$.

Définition 1 ***(Skyline)*** *Le skyline de l'ensemble de données E sur l'espace D, avec $\wp$ l'ensemble des préférences associées à D, est l'ensemble des points qui ne sont dominés par aucun autre point dans E : $Sky(D,E)_{\wp} = \{p \in E | \ (\forall q \in E, \neg(q \prec_D p))\}$.*

Propriété 1 *Bouadi et al. (2013)* ***(Monotonie de l'extension d'une préférence)*** *Soient $\wp'$ et $\wp''$ deux ensembles de préférences sur D. Si $\wp' \subseteq \wp''$ (i.e. $\wp''$ est une extension de $\wp'$) alors, $Sky(D,E)_{\wp''} \subseteq Sky(D,E)_{\wp'}$.*

ID parcelle	Loc	Sn	Re
a	Pays de la Loire (PL)	16	200
b	Yeres (YRS)	24	500
c	Vilaine (VLN)	36	100
d	Yar	30	200
e	ALL	23	400
f	Epte (EPT)	30	300

TAB. 1: Ensemble de parcelles

Les différentes définitions dans cette section sont illustrées à partir de l'exemple de la table 1.

Exemple 1 *L'exemple de la table 1 contient un ensemble de 6 parcelles agricoles décrites par 3 dimensions : la localisation (Loc), le rendement (Re) kg/ha et le taux de pollution nitrique (Sn) kgN/ha/année. Une préférence de base $\wp^0_{d_i}$ est définie pour chaque dimension d_i de l'espace de dimensions D. Les relations d'ordre $\leqslant_{Re}$ et $\leqslant_{Sn}$ s'appuient sur la relation d'ordre sur les naturels, et indiquent qu'on préfère les parcelles ayant le rendement Re le plus élevé et le taux de pollution nitrique Sn le moins élevé. Les valeurs de la dimension Loc sont associées à la préférence de base suivante : $Bretagne <_{Loc} Epte$ et $Yeres <_{Loc} Normandie$. Les autres valeurs de la dimension Loc sont laissées non ordonnées. $Sky(D,E)_{\wp^0} = \{a,b,c,d,e,f\}$.*

Les préférences définies sur l'espace D sont représentées par un ensemble de préférences binaires (ex. la préférence de la dimension Loc est composée de deux préférences binaires : $((Bretagne, Epte), (Yeres, Normandie))$. Si le domaine des valeurs d'une dimension donnée est un ensemble indénombrable (ex. l'ensemble des réels) et totalement ordonné, alors l'ensemble des préférences binaires associées à ces valeurs ne peut être totalement listé. Par exemple, dans l'exemple de la table 1, l'ensemble des préférences binaires associées à la dimension Re ne peut être listé de façon exhaustive, et sera noté dans la suite par $\wp^0_{Re}$.

2.1 Formalisation des hiérarchies

Dans les entrepôts de données, le domaine de valeurs des dimensions est généralement structuré sous forme de hiérarchies. La notion de hiérarchie de dimension est

Définition 2 ***(Hiérarchie de dimension)*** *Une hiérarchie sur la dimension $d_i \in D$ est représentée par un graphe orienté acyclique dans lequel les noeuds représentent les valeurs du domaine de d_i, et les arcs représentent la relation d'inclusion ensembliste. La valeur la plus générale correspond au noeud le plus élevé de la hiérarchie (i.e. racine) et les feuilles correspondent aux valeurs les plus spécifiques.*

Une hiérarchie est associée à chaque dimension de l'espace de dimensions D.

Définition 3 *Soit $H_D = \{h_{d_1}, ..., h_{d_{|D|}}\}$ l'ensemble des hiérarchies associées aux dimensions de l'espace D.*

- *h_{d_i} représente la relation hiérarchique entre les valeurs de la dimension $d_i \in D$,*
- *h_{d_i} est un graphe orienté acyclique,*
- *Pour chaque noeud n_j de h_{d_i} , $label(n_j) = v_j$ avec $v_j \in dom(d_i)$,*
- *Pour chaque valeur $v_j \in dom(d_i)$, $\exists n_j \in h_{d_i}$, $label(n_j) = v_j$.*

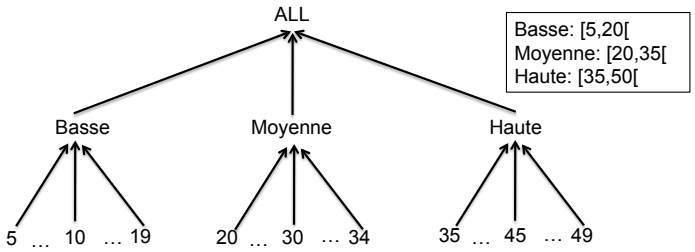
$\widehat{v_j}$ désigne l'ensemble des labels des ancêtres (directs ou indirects) de n_j dans la hiérarchie h_{d_i}. $\check{v_j}$ désigne l'ensemble des labels des descendants de n_j.
Nous imposons que, pour chaque dimension d_i de l'espace D, la valeur ALL appartienne au domaine de valeurs de d_i. La valeur ALL correspond à la valeur la plus générale dans la hiérarchie d'une dimension quelconque.

Exemple 2 *La Figure 1 donne des exemples de hiérarchies définies sur les dimensions Loc et Sn. La structure hiérarchique de la dimension Loc est représentée par trois niveaux hiérarchiques : la région (Bretagne, Normandie,...), la sous-région (Haute Bretagne, Basse Normandie,...) et le bassin versant (Yar, Epte,...). Toutes les valeurs associées à ces niveaux hiérarchiques appartiennent au domaine de la dimension Loc. De même, la hiérarchie sur la dimension Sn introduit des valeurs qui catégorisent des plages de valeurs numériques associées à cette dimension.*

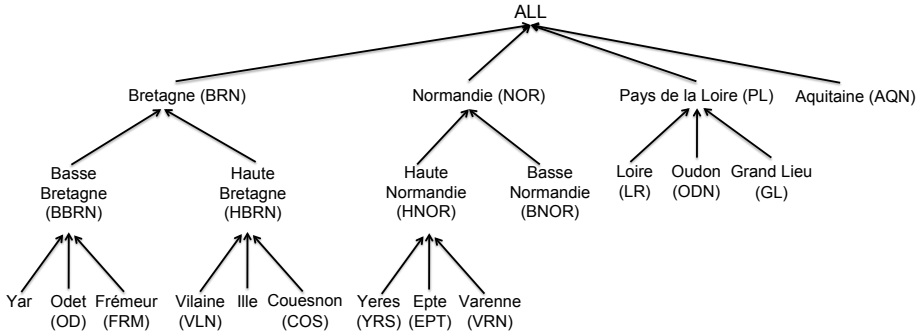
L'existence d'une hiérarchie permet de représenter des informations incomplètes. Par exemple, la parcelle e de la table 1 est décrite par la valeur ALL sur la dimension Loc. Cela signifie qu'il n'y a pas d'information sur l'endroit exact où la parcelle e se situe. De la même façon, la parcelle a de la table 1 est décrite par la valeur *Pays de Loire* sur la dimension Loc. Cela signifie que la valeur la plus détaillée et la plus précise dont nous disposons concernant la localisation de la parcelle a est sa région (*Pays de Loire*). Pour les parcelles b, c, d et f, nous disposons des valeurs les plus précises du domaine de la dimension Loc.

La définition de hiérarchies sur les dimensions permet à l'utilisateur de spécialiser ou de généraliser ses préférences lors du calcul des skyline. Prenons l'exemple de la dimension Sn décrite dans la table 1, nous remarquons qu'avant la structuration des valeurs de cette dimension en hiérarchie, l'utilisateur pouvait seulement formuler des préférences entre les valeurs individuelles (ex. une parcelle avec un taux de pollution nitrique de 16 est meilleure qu'une parcelle avec un taux de 30). Imaginons maintenant que l'utilisateur s'intéresse à des taux de pollution regroupés selon leur impact environnemental (ex. les taux inférieurs à 20 (i.e. bas) ont un faible impact, les taux entre 20 et 34 (i.e. moyens) ont un impact moyen et les taux supérieurs à 35 (i.e. hauts) ont un fort impact). L'utilisateur souhaite formuler des préférences entre ces intervalles de valeurs. La définition d'une hiérarchie sur la dimension Sn permet la formulation de telles préférences et permet à l'utilisateur de poser des requêtes à différents niveaux

hiérarchiques de la dimension. De la même manière, un utilisateur qui s'intéresse à l'étude du phénomène de proliférations d'algues vertes dans les bassins versants agricoles côtiers est susceptible de porter plus d'attention à la région de Bretagne (dimension Loc) qu'aux autres régions. L'utilisateur souhaitera alors accéder à un niveau plus précis des données relatives à la région de Bretagne afin de formuler ses préférences et extraire les points skyline associés. Il est alors intéressant de fournir à l'utilisateur un moyen pour spécialiser ses préférences à un niveau plus détaillé que la région (ex. niveau du bassin versant). La définition d'une hiérarchie sur la dimension Loc permet de telles opérations.



(a) La hiérarchie de la dimension taux de pollution nitrique Sn



(b) La hiérarchie de la dimension Localisation Loc

FIG. 1

2.2 Relations hiérarchiques entre préférences

Dans cette sous-section, nous nous focalisons sur les nouvelles propriétés des préférences introduites par les dimensions hiérarchiques.

Définition 4 ***(Consistance d'une préférence)***
Soit $\wp_{d_i} = \{(v_1, v_2), ..., (v_n, v_m)\}$ *une préférence sur la dimension hiérarchique* d_i.
La préférence $\wp_{d_i}$ *est consistante si et seulement si elle ne contient pas de préférence binaire ordonnant une valeur et un de ses ancêtres :* $\nexists (v_k, v_j) \in \wp_{d_i}, k \neq j, (v_k \in \widehat{v_j} \vee v_j \in \widehat{v_k})$.

Dans la suite de cet article, nous considérons seulement des préférences consistantes.

Définition 5 ***(Fermeture hiérarchique)***
Soit $\wp_{d_i}$ une préférence associée à la dimension hiérarchique d_i de l'espace de dimensions D. La fermeture hiérarchique de la préférence $\wp_{d_i}$, notée $(\wp_{d_i})^H$, est définie par l'ensemble des préférences binaires obtenues par fermeture transitive sur les descendants des valeurs de $\wp_{d_i}$:
$(\wp_{d_i})^H = \{(v_p, v_q) \mid \exists\ (v_n, v_m) \in \wp_{d_i}, (v_p \in \widetilde{v_n} \vee v_p = v_n) \wedge (v_q \in \widetilde{v_m} \vee v_q = v_m)\}$.

Exemple 3 *Soit $\wp_{Loc} = \{(BRN, EPT)\}$.*
La fermeture hiérarchique de $\wp_{Loc}$ est :
$(\wp_{Loc})^H = \{(BRN, EPT), (BBRN, EPT), (HBRN, EPT), (Yar, EPT), (OD, EPT), (FRM, EPT), (Ille, EPT), (VLN, EPT), (COS, EPT)\}$.

Propriété 2 *La fermeture hiérarchique d'une préférence $\wp$ est consistante ssi $\wp$ est consistante.*

Dans la suite, nous ne considérons que les fermetures hiérarchiques des préférences.

Partant d'une préférence initiale $\wp^0$, appelée préférence de base, nous voulons permettre à l'utilisateur de naviguer dans les hiérarchies des dimensions associées. Cela revient à spécialiser ou à généraliser les valeurs ordonnées dans la préférence de base en suivant les liens hiérarchiques. La spécialisation d'une valeur d'une préférence correspond à l'ajout d'un ordre (total ou partiel) sur tous les descendants directs de la valeur en question. Ces descendants doivent être initialement non ordonnés. Si un ordre est déjà défini sur une partie ou sur la totalité de ses descendants, la complétion au même niveau de détail de cet ordre n'est pas considérée comme une spécialisation mais comme une extension (cf. propriété 1). La spécialisation revient à lever de l'indifférence. Il y a deux façons de le faire. Introduire une nouvelle préférence soit sur des valeurs citées explicitement dans la préférence de base, soit sur des valeurs ignorées dans la préférence de base. Ceci nous permet de désigner l'ensemble des valeurs du domaine de Loc qui peuvent être spécialisées (cf. les valeurs en vert (i.e. valeurs explicitement ordonnées) et en bleu (i.e. valeurs non ordonnées) de la Figure 2) et celles qui ne peuvent pas l'être (cf. les valeurs en rouge de la Figure 2). Ceci permet de tracer une frontière entre ces deux types de valeurs (cf. Figure 2).

Exemple 4 *Soient $\wp^0_{Loc}$ et $\wp'_{Loc}$ deux préférences associées à la dimension hiérarchique Loc.*
$\wp^0_{Loc} = (\{(BRN, EPT)\})^H = \{(BRN, EPT), (BBRN, EPT), (HBRN, EPT), (Yar, EPT), (OD, EPT), (FRM, EPT), (Ille, EPT), (VLN, EPT), (COS, EPT)\}$,
$\wp'_{Loc} = \wp^0_{Loc} \bigcup \{(Yar, VLN)\}$ *et* $\wp''_{Loc} = \wp^0_{Loc} \bigcup \{(LR, GL)\}$.
$\wp'_{Loc}$ *est une spécialisation de* $\wp^0_{Loc}$ *car les valeurs de* $\{(Yar, VLN)\}$ *sont des descendants de* BRN *dans* h_{Loc}.
$\wp''_{Loc}$ *est une spécialisation de* $\wp^0_{Loc}$ *par rapport à* h_{Loc}. *En effet, l'ancêtre de* GL *et* LR *(i.e.* PL*) n'apparaît pas dans la préférence* $\wp^0_{Loc}$. *Cela signifie que la valeur* PL *n'est pas ordonnée explicitement dans* $\wp^0_{Loc}$ *par rapport aux autres valeurs (i.e.* BRN *et* NOR*).*
Considérons maintenant l'exemple suivant :
$\wp^0_{Loc} = (\{(BRN, EPT)\})^H$ *et* $\wp'_{Loc} = \wp^0_{Loc} \bigcup \{(VLN, YRS)\}$.
$\wp'_{Loc}$ *est une extension de* $\wp^0_{Loc}$ *mais* $\wp'_{Loc}$ *n'est pas une spécialisation de* $\wp^0_{Loc}$. *En effet, la valeur* YRS *citée dans* $\wp'_{Loc}$ *n'est la spécialisation d'aucune des valeurs citées dans* $\wp^0_{Loc}$ *(i.e. valeurs en vert ou en bleu dans la Figure 2).*

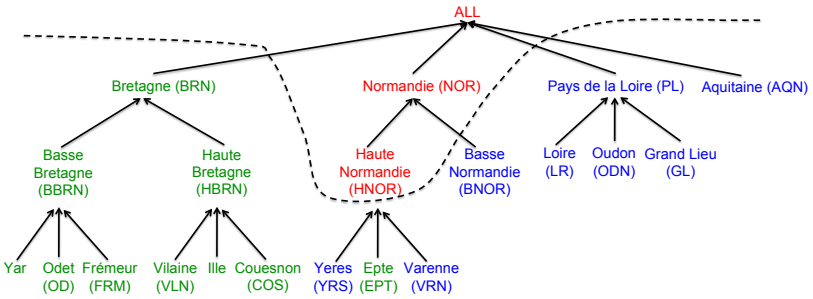


FIG. 2: Frontière implicite de spécialisation/généralisation

La généralisation d'une préférence est l'opération duale de la spécialisation d'une préférence. Elle consiste en l'ajout de l'indifférence entre des valeurs explicitement ou implicitement ordonnées de la préférence en question. Autrement dit, il s'agit de relaxer un ordre exprimé sur les descendants d'une valeur citée dans la préférence en question.

Exemple 5 *Soient* $\wp^0_{Loc} = (\{(BRN, EPT)\})^H$ *et* $\wp^{'}_{Loc} = \wp^0_{Loc} \bigcup \{(Yar, VLN)\}$ *deux préférences associées à la dimension* Loc.
$\wp^0_{Loc}$ *est une généralisation de* $\wp^{'}_{Loc}$ *par rapport à* h_{Loc}, *car* $\widehat{Yar} = \widehat{VLN} = BRN$.
Prenons un autre exemple avec $\wp^0_{Loc}$ *et* $\wp^{''}_{Loc} = \wp^0_{Loc} \bigcup \{(LR, GL)\}$.
$\wp^0_{Loc}$ *est une généralisation de* $\wp^{''}_{Loc}$ *par rapport à* h_{Loc}. *En effet, l'ancêtre de* GL *et* ODN *(i.e.* PL*) n'apparaît pas dans la préférence* $\wp^0_{Loc}$. *Cela signifie que la valeur* PL *n'est pas ordonnée dans* $\wp^0_{Loc}$ *par rapport aux autres valeurs (i.e.* BRN *et* NOR*).*

La définition 6 définit formellement les opérations de spécialisation/généralisation :

Définition 6 ***(Spécialisation/Généralisation d'une préférence)***
Soient $\wp_{d_i} = \{(v_1, v_2), ..., (v_n, v_m)\}$ *et* $\wp'_{d_i} = \{(v'_1, v'_2), ..., (v'_p, v'_q)\}$ *deux préférences associées à la dimension hiérarchique* d_i.

- $\wp'_{d_i}$ *est une spécialisation de* $\wp_{d_i}$ *(notée* $\wp_{d_i} \subset_h \wp'_{d_i}$*) si :*
  1. $\wp_{d_i} \subset \wp'_{d_i}$ *i.e.* $\wp'_{d_i}$ *est une extension de* $\wp_{d_i}$,
  2. $\forall (v'_i, v'_j) \in \wp'_{d_i}$:
     - *soit (a) :* $(v'_i, v'_j) \in \wp_{d_i}$,
     - *soit (b) :*
       - $\exists (v_k, v_m) \in \wp_{d_i}$, $(v'_i \in \check{v_k} \wedge v'_j \in \check{v_k}) \vee (v'_i \in \widetilde{v_m} \wedge v'_j \in \widetilde{v_m})$ *ou,*
       - $\widehat{v'_i} \cap \widehat{v'_j} \neq \varnothing \wedge \forall v \in \widehat{v'_i} \cap \widehat{v'_j}$, v *n'est pas ordonnée dans* $\wp_{d_i}$.

     Et il existe au moins un couple (v'_i, v'_j) *qui vérifie la propriété (b).*
- $\wp_{d_i}$ *est une généralisation de* $\wp'_{d_i}$ *si* $\wp'_{d_i}$ *est une spécialisation de* $\wp_{d_i}$.
- $\wp_{d_i}$ *et* $\wp'_{d_i}$ *sont incomparables s'il n'existe aucune relation de spécialisation ou de généralisation entre elles* ($\neg(\wp_{d_i} \subset_h \wp'_{d_i}) \wedge \neg(\wp'_{d_i} \subset_h \wp_{d_i})$),

Nous étendons la définition de spécialisation/généralisation d'une préférence à plusieurs dimensions hiérarchiques.

Définition 7 *Soient $\wp = \bigcup_{d_i \in D} \wp_{d_i}$ et $\wp' = \bigcup_{d_i \in D} \wp'_{d_i}$ deux préférences associées à l'espace de dimensions D. $\wp'$ est une spécialisation de $\wp$ (i.e. $\wp$ est une généralisation de $\wp'$) ssi : $\forall d_i \in D$, $\wp_{d_i} \subset_h \wp'_{d_i} \vee \wp_{d_i} = \wp'_{d_i} \wedge \exists d_i \in D$, $\wp_{d_i} \subset_h \wp'_{d_i}$.*

Toutes les spécialisations et généralisations d'une préférence associée à un ensemble de dimensions constituent un ensemble partiellement ordonné où l'ordre est déterminé par la relation de spécialisation/généralisation. Par exemple, toutes les spécialisations et généralisations de la préférence de base $\wp^0$ définie sur les dimensions de la table 1, induisent la structure hiérarchique de la Figure 3. Chaque noeud de cette structure est associé à une spécialisation ou à une généralisation de la préférence de base $\wp^0$. La notation $\wp^{kl}$, avec k un numéro de niveau dans la structure hiérarchique et l un numéro séquentiel associé à un noeud de niveau k, est parfois utilisée dans certaines figures afin de les rendre plus lisibles. Soit un noeud quelconque de la structure hiérarchique $HSky$ associé à une préférence $\wp$. Les préférences associées aux descendants directs (resp. ancêtres directs) de ce noeud sont appelées spécialisations directes (resp. généralisations directes) de $\wp$.

3 Requêtes skyline associées à des dimensions hiérarchiques

D'après la définition 6, la spécialisation d'une préférence constitue une extension de cette dernière, mais l'inverse n'est pas vérifié. Nous pouvons alors déduire un corollaire de la propriété 1 de monotonie de l'extension d'une préférence.

Corollaire 1 ***(Monotonie hiérarchique)*** *Soient $\wp$ et $\wp'$ deux préférences sur l'espace de dimensions D. Si $\wp'$ est une spécialisation de $\wp$ ($\wp \subset_h \wp'$), alors $\wp'$ est aussi une extension de $\wp$, et par conséquent, $Sky(D,E)_{\wp'} \subseteq Sky(D,E)_{\wp}$.*

Exemple 6 *Soient $\wp^0 = (\{(BRN, EPT)\})^H \bigcup \wp^0_{Sn} \bigcup \wp^0_{Re}$ la préférence de base définie sur l'espace D, et $\wp' = \wp^0_{Loc} \bigcup \{(Yar, VLN)\}\ \wp^0_{Sn} \bigcup \wp^0_{Re}$ une préférence définie sur ce même espace. $\wp'$ est une spécialisation de $\wp^0$. $Sky(D,E)_{\wp^0} = \{a,b,e,c,d,f\}$ et $Sky(D,E)_{\wp'} = \{a,b,d,e,f\}$. Nous avons bien $Sky(D,E)_{\wp'} \subseteq Sky(D,E)_{\wp^0}$.*

La propriété de monotonie hiérarchique indique que tout point skyline associé à une préférence donnée reste skyline lorsque nous considérons une généralisation de cette préférence. Cette propriété va faciliter considérablement l'analyse et le calcul des points skyline en présence de préférences associées à des dimensions hiérarchiques.

Dans ce qui suit, nous désignerons par préférences hiérarchiques d'une préférence $\wp$, les spécialisations et généralisations de cette dernière.

3.1 Requêtes skyline hiérarchiques : Algorithme *HSky*

Notre objectif est de minimiser le nombre de tests de dominance pour calculer efficacement les skyline lors de la navigation sur les dimensions hiérarchiques. Nous souhaitons donc caractériser les points skyline qui restent skyline et ceux qui ne le sont plus ou qui le deviennent, lors de la spécialisation (drill-down) ou de la généralisation (roll-up) des préférences associées aux dimensions hiérarchiques. Nous proposons un compromis entre (i) *matérialiser* tous les points

skyline de toutes les préférences hiérarchiques associées à la préférence de base $\wp^0$, et (ii) *calculer*, pour chaque requête utilisateur, les points skyline associés aux préférences hiérarchiques formulées dans la requête.

3.1.1 Calcul des requêtes skyline hiérarchiques

Dans le but de calculer les points skyline pour une requête utilisateur associée à une spécialisation quelconque $\wp'$ de la préférence $\wp$, nous introduisons l'ensemble de points *hierarchical skyline* $HNSky(D,E)_{(\wp',\wp)}$, où $\wp$ est un ancêtre direct de $\wp'$ dans la structure de spécialisation/généralisation. Cet ensemble regroupe les points qui sont disqualifiées lors de la spécialisation de $\wp$ en $\wp'$ et donc les points skyline introduits par la généralisation de $\wp'$ en $\wp$.

Définition 8 ***(HNSky : Hierarchical New Skyline)*** *Soient $\wp$ et $\wp'$ deux préférences associées à D avec $\wp'$ une spécialisation directe de $\wp$, $Sky(D,E)_{\wp'}$ le skyline associé à $\wp'$ et $Sky(D,E)_{\wp}$ le skyline associé à $\wp$. L'ensemble des points de $HNSky_{(\wp,\wp')}$ est défini par :*
$HNSky(D,E)_{(\wp,\wp')} = \{p \in Sky(D,E)_{\wp} | p \notin Sky(D,E)_{\wp'}\}$.
Nous avons donc : $HNSky(D,E)_{(\wp,\wp')} = Sky(D,E)_{\wp} - Sky(D,E)_{\wp'}$.

Exemple 7 *Soient* $\wp = (\{(BRN, EPT)\})^H \bigcup \wp_{Sn} \bigcup \wp^0_{Re}$ *et*
$\wp' = \wp \bigcup \{(Yar, VLN)\}$ *deux préférences définies sur ce l'espace D.*
$\wp'$ est une spécialisation directe de $\wp^0$.
$Sky(D,E)_{\wp'} = Sky(D,E)_{\wp} - HNSky(D,E)_{(\wp,\wp')} = \{a,b,c,d,e,f\} - \{c\} = \{a,b,d,e,f\}$.

Nous souhaitons construire une structure $HSky$ pour stocker efficacement toutes les informations pré-calculées. Notre objectif est d'éviter de stocker tous les points skyline associés à chaque préférence hiérarchique possible sur l'espace de dimensions D. Pour chaque arc, de la structure $HSky$, connectant un noeud fils associé à la préférence $\wp'$ à un noeud parent associé à la préférence $\wp$, nous calculons et stockons l'ensemble $HNSky(D,E)_{(\wp,\wp')}$.
Tout d'abord, nous fixons la préférence de base $\wp^0$. À partir de celle-ci, nous allons naviguer dans ses spécialisations ou ses généralisations. Ensuite, la construction de la structure $HSky$ se déroule en deux étapes :

- Génération de toutes les spécialisations et généralisations directes et indirectes de $\wp^0$ et construction de la hiérarchie des préférences qui donne la structure de $HSky$. Le noeud au sommet de la structure de spécialisations/généralisations correspond à la préférence $\varnothing$ sur les dimensions hiérarchiques,
- Pour chaque arc de $HSky$, calcul de l'ensemble $HNSky$ enregistrant les points disqualifiés (resp. réintroduits) en passant d'une préférence à une spécialisation directe de cette préférence (resp. d'une préférence à une généralisation directe de cette préférence).

Les ensembles $HNSky$ sont générés de manière *Bottom-Up* pour les généralisations de $\wp^0$, et de manière *Top-Down* pour les spécialisations de $\wp^0$. En effet, grâce à la propriété 1, nous avons montré que lors d'une spécialisation ou d'une généralisation d'une préférence, certains des points skyline associés à la nouvelle préférence peuvent être dérivés sans test de dominance. Nous pouvons aussi déduire que chaque point de l'ensemble $Sky(D,E)_{\wp^0}$ associé au noeud $\wp^0$, appartient à l'ensemble skyline associé à chacun de ses noeuds parents (généralisation de $\wp^0$). Par conséquent, pour calculer l'ensemble $HNSky$ d'un noeud parent (généralisation de $\wp^0$), il suffit de tester la dominance des points qui n'appartiennent pas à l'ensemble $Sky(D,E)_{\wp^0}$ et aux ensembles $HNSky$ associés aux noeuds fils.

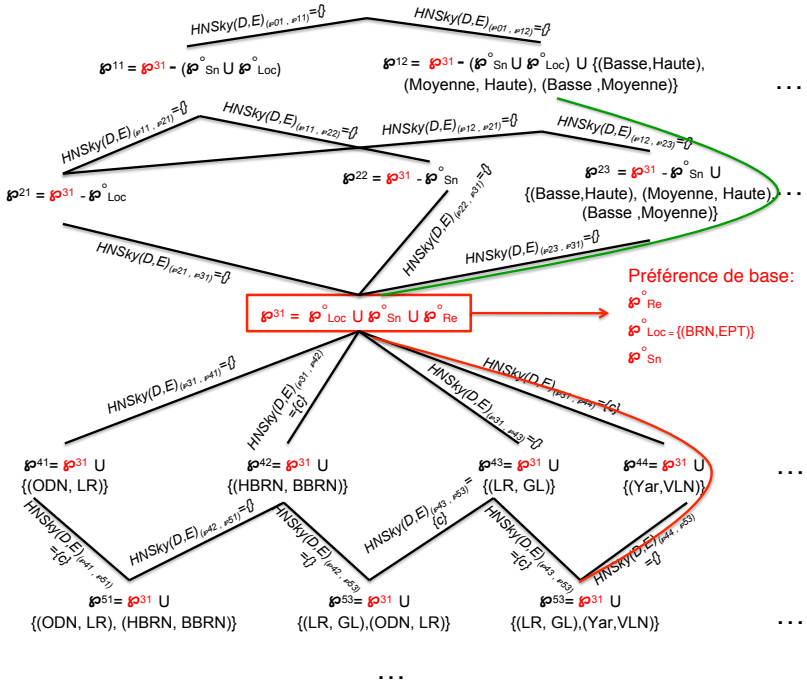


FIG. 3: Structure de spécialisation/généralisation $HSky$

Pour calculer l'ensemble $HNSky$ associé à toute spécialisation de la préférence $\wp^0$, nous devons seulement tester les points appartenant à $Sky(D,E)_{\wp^0}$. En effet, grâce à la propriété 1, nous savons que le skyline associé à une spécialisation de $\wp^0$ est soit égal soit inclus dans $Sky(D,E)_{\wp^0}$. La structure $HSky$ obtenue est présentée dans la Figure 3. En pratique, la génération des spécialisations / généralisations et le calcul des $HNSky$ associés sont effectués simultanément.

3.1.2 Évaluation de requêtes

Dans l'analyse en ligne traditionnelle OLAP, les opérations *drill-down* et *roll-up* sont appliquées sur les hiérarchies des dimensions. Dans notre approche, ces opérations sont appliquées sur des préférences hiérarchiques associées à des requêtes skyline. Le stockage dans la structure $HSky$ nous permet de réduire le temps de calcul des points skyline en présence de préférences hiérarchiques et d'assurer une meilleure interactivité. Dans la suite, nous utilisons la structure $HSky$ décrite dans la Figure 3 pour illustrer la navigation à partir du skyline de la préférence de base $\wp^0 = (\{(BRN, EPT)\})^H \bigcup \wp^0_{Sn} \bigcup \wp^0_{Re}$. Le skyline relatif à $\wp^0$ est $Sky(D,E)_{\wp^0} = \{a, b, e, c, d, f\}$. Nous détaillons comment calculer, en utilisant la structure $HSky$, les points skyline associés aux préférences $\wp' = \wp^0 \bigcup \{(LR,GL),(Yar,VLN)\}$ (une spécialisation de $\wp^0$), et $\wp = \{(Moyenne,Haute),(Basse,Moyenne), (Basse,Haute)\} \bigcup \wp^0_{Re}$ (une généralisation de $\wp^0$).

Algorithme 1 : $HSky(\wp^0, \wp, Sky(D,E)_{\wp^0})$

input : $\wp^0$: la préférence de base associée au noeud origine, $Sky(D,E)_{\wp^0}$: skyline associé à la préférence $\wp^0$, $\wp$: préférence associée au noeud cible
output : $Sky(D,E)_{\wp}$: skyline associé à $\wp$

$Sky \leftarrow Sky(D,E)_{\wp^0}$;
si $\wp \subset_h \wp^0$// Tester si $\wp$ est une généralisation de $\wp^0$;
alors
  pour chaque *noeud parent* $\wp^1$ *de* $\wp^0$ **faire**
    si $\wp = \wp^1$ **alors**
      $Sky \leftarrow Sky(D,E)_{\wp^0} \bigcup HNSky(D,E)_{(\wp^1,\wp^0)}$;
      $Sky(D,E)_{\wp} \leftarrow Sky$;
    sinon
      si $\wp \subset_h \wp^1$// Tester si $\wp$ est une généralisation de $\wp^1$;
      alors
        $Sky \leftarrow Sky(D,E)_{\wp^0} \bigcup HNSky(D,E)_{(\wp^1,\wp^0)}$;
        $HSky(\wp^1, \wp, Sky)$;
        Sortir;
si $\wp^0 \subset_h \wp$// Tester si $\wp$ est une spécialisation de $\wp^0$;
alors
  pour chaque *noeud fils* $\wp^1$ *de* $\wp^0$ **faire**
    si $\wp = \wp^1$ **alors**
      $Sky \leftarrow Sky(D,E)_{\wp^0}$ - $HNSky(D,E)_{(\wp^0,\wp^1)}$;
      $Sky(D,E)_{\wp} \leftarrow Sky$;
    sinon
      si $\wp^1 \subset_h \wp$// Tester si $\wp$ est une spécialisation de $\wp^1$;
      alors
        $Sky \leftarrow Sky(D,E)_{\wp^0}$ - $HNSky(D,E)_{(\wp^0,\wp^1)}$;
        $HSky(\wp^1, \wp, Sky)$;
        Sortir;
Retourner $Sky(D,E)_{\wp}$;

Skyline d'une préférence spécialisée Le skyline associé à la préférence $\wp'$, spécialisation de la préférence de base $\wp^0$, est calculé comme suit.
L'exploration commence à partir du noeud associé à la préférence de base $\wp^0$ et explore récursivement les noeuds fils en profondeur, à la recherche du noeud associé à la préférence $\wp'$ selon l'algorithme 1. Lorsque le noeud recherché est atteint, $Sky(D,E)_{\wp'}$ est calculé en retirant de

l'ensemble $Sky(D,E)_{\wp^0}$ l'union des ensembles $HNSky$ de chaque arc du chemin menant de $\wp^0$ à $\wp'$ (Algorithme 1).

Pour plus d'efficacité, la recherche n'explore que les noeuds associés à des généralisations de la préférence $\wp'$ (Algorithme 1). Cet élagage assure que le chemin parcouru entre le noeud associé à $\wp^0$ et celui associé à $\wp'$ est de taille minimale.

Exemple 8 $\wp^0 = (\{(BRN, EPT)\})^H \bigcup \wp^0_{Sn} \bigcup \wp^0_{Re}$. *Le skyline relatif à* $\wp^0$ *est* $Sky(D,E)_{\wp^0}$ $= \{a,b,e,c,d,f\}$. *Soit* $\wp' = \wp^0 \bigcup \{(LR,GL), (Yar,VLN)\}$ *une spécialisation de* $\wp^0$.
Dans la Figure 3, $\wp^0$ *correspond à* $\wp^{31}$ *et* $\wp'$ *correspond à* $\wp^{53}$. *Le chemin parcouru pour calculer le skyline associé à* $\wp'$ *est indiqué en rouge.*
$Sky(D,E)_{\wp'} = Sky(D,E)_{\wp^{31}}$ - $(HNSky(D,E)_{(\wp^{31},\wp^{44})} \bigcup HNSky(D,E)_{(\wp^{44},\wp^{53})})$
$= \{a,b,c,e,d,f\}$ -($\{c\} \bigcup \{\}$) $= \{a,b,e,d,f\}$.

Skyline d'une préférence généralisée Le skyline associé à la préférence $\wp$, généralisation de la préférence de base $\wp^0$, est calculé selon un processus symétrique.
L'exploration commence avec comme noeud courant le noeud associé à la préférence de base $\wp^0$ et explore récursivement les noeuds parents en profondeur, à la recherche du noeud associé à la préférence $\wp$ selon l'algorithme 1. Lorsque le noeud recherché est atteint, $Sky(D,E)_{\wp}$ est calculé en réalisant l'union des ensembles $HNSky$ de chaque arc du chemin menant de $\wp^0$ à $\wp'$ union l'ensemble $Sky(D,E)_{\wp^0}$ (Algorithme 1).

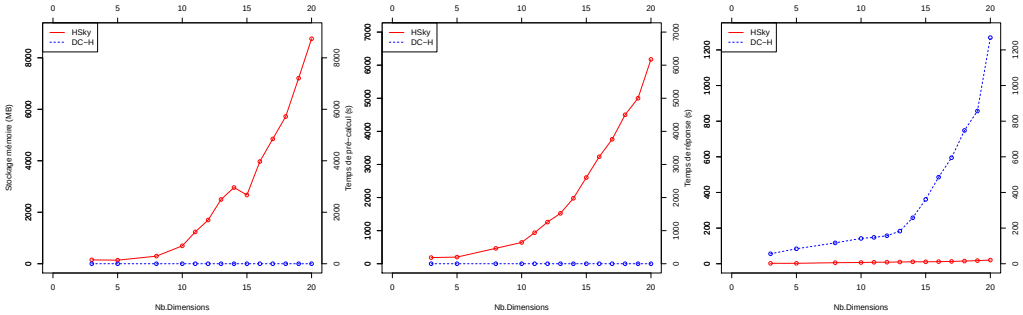
Pour plus d'efficacité, la recherche n'explore que les noeuds associés à des spécialisations de la préférence $\wp$ (Algorithme 1). Cet élagage assure que le chemin parcouru entre le noeud associé à $\wp^0$ et celui associé à $\wp$ est de taille minimale.

Exemple 9 $\wp^0 = (\{(BRN, EPT)\})^H \bigcup \wp^0_{Sn} \bigcup \wp^0_{Re}$. *Le skyline relatif à* $\wp^0$ *est*
$Sky(D,E)_{\wp^0} = \{a,b,e,c,d,f\}$.
Soit $\wp = \{(Moyenne,Haute), (Basse,Moyenne), (Basse,Haute)\} \bigcup \wp^0_{Re}$ *une généralisation de* $\wp^0$. *Dans la Figure 3,* $\wp^0$ *correspond à* $\wp^{31}$ *et* $\wp$ *correspond à* $\wp^{12}$. *Le chemin parcouru pour calculer le skyline associé à* $\wp$ *est indiqué en vert.*
$Sky(D,E)_{\wp} = Sky(D,E)_{\wp^{31}} \bigcup HNSky(D,E)_{(\wp^{23},\wp^{31})} \bigcup HNSky(D,E)_{(\wp^{12},\wp^{23})}$
$= \{a,b,c,e,d,f\} \bigcup \{\} \bigcup \{\} = \{a,b,c,d,e,f\}$.

4 Évaluations expérimentales

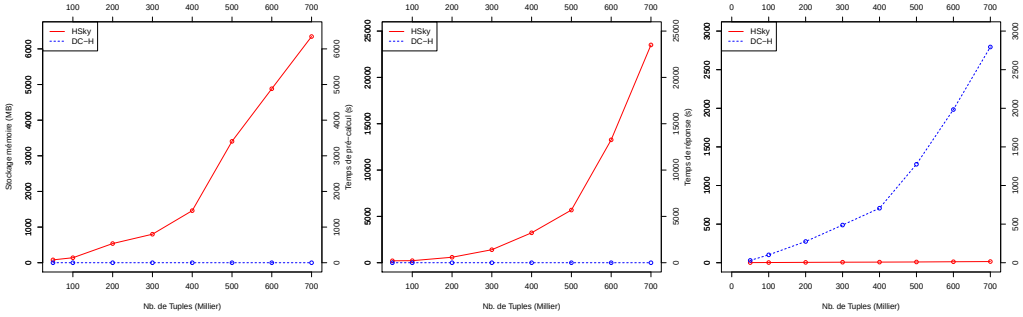
Dans cette section, nous présentons une évaluation expérimentale de notre algorithme $HSky$ sur des données synthétiques. L'algorithme $HSky$ a été implémenté en langage $JAVA$. Nous avons conduit nos expérimentations sur un Intel Xeon CPU 3GHz et 16 GB de RAM sous une plateforme Linux. Pour les dimensions à un seul niveau hiérarchique, les données ont été produites par le générateur publié par les auteurs de Borzsonyi et al. (2001). Trois types de jeux de données ont été générés : des données indépendantes, des données corrélées et des données non-corrélées. L'article de Borzsonyi et al. (2001) fournit la description de ces jeux de données. Nous présentons seulement les résultats relatifs aux données non-corrélées. Les résultats relatifs aux données indépendantes et aux données corrélées sont similaires, mais les temps de pré-calcul et de réponse aux requêtes sont beaucoup plus courts pour les données corrélées. Les dimensions hiérarchiques ont été générées selon une distribution de Zipfian Trenkler

(1994). Par défaut, nous avons initialisé le paramètre Zipfian θ à 1 (données non-corrélées). Nous avons obtenus 700.000 tuples avec 6 dimensions à un seul niveau hiérarchique. Nous avons fait varier le nombre de dimensions hiérarchiques de 3 à 20 et le nombre de niveaux hiérarchiques de ces dimensions de 3 à 7. Nous avons choisi une préférence de base de sorte que le nombre de spécialisations soit équilibré avec le nombre de généralisations. Le template de la préférence posée dans la requête a été choisi de sorte que cette préférence représente une spécialisation / généralisation indirecte de la préférence de base.



(a) (b) (c)

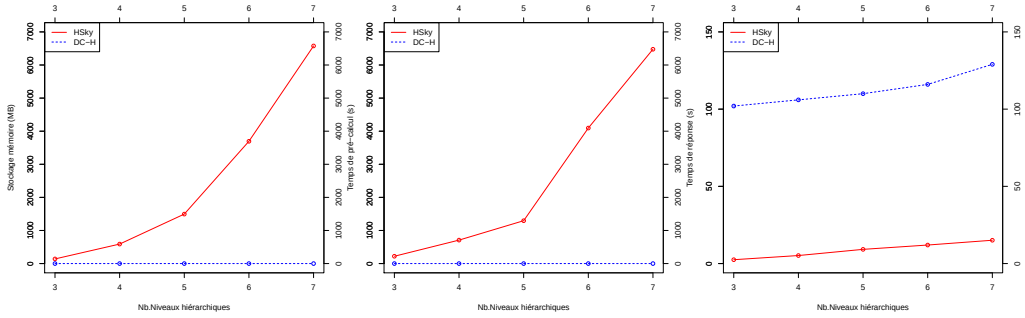
FIG. 4: Variation du nombre de dimensions hiérarchiques



(a) (b) (c)

FIG. 5: Variation de la taille de l'ensemble de données

À notre connaissance, il n'existe pas de travaux dans la littérature conciliant extraction de points skyline et prise en compte de dimensions hiérarchiques. Nous avons alors implémenté l'algorithme DC-H, basée sur le principe de l'algorithme de calcul de skyline $Divide$ & $Conquer$ ($D\&C$) Borzsonyi et al. (2001). DC-H ne matérialise aucun résultat partiel, il recalcule à chaque niveau hiérarchique l'ensemble des skyline. Étant donné que $D\&C$ est une façon naturelle d'exprimer des algorithmes parallèles, et afin d'améliorer les performances de



(a) (b) (c)

FIG. 6: Variation du nombre de niveaux d'une dimension hiérarchique

DC-H, nous avons parallélisé le calcul de DC-H. L'algorithme $HSky$ est basé sur le principe de DC-H pour le calcul des skyline. Dans ces expérimentations, nous comparons les performances de l'algorithme $HSky$ avec l'algorithme DC-H en illustrant :

a) La place mémoire qui correspond pour $HSky$ au stockage de la structure associée,

b) Le temps de précalcul qui correspond pour HSy à la construction de la structure $HSky$,

c) Le temps de réponse à une requête skyline avec une préférence qui est une spécialisation/généralisation de la préférence de base.

Variation du nombre de dimensions hiérarchiques Dans cette première expérimentation, le nombre de dimensions à un seul niveau hiérarchique est fixé à 6 et le nombre de dimensions hiérarchiques varie de 3 à 20. La Figure 4 montre que la taille mémoire et le temps de pré-calcul de $HSky$ augmentent avec le nombre de dimensions hiérarchiques. Cela est dû à la complexité de la structure $HSky$ qui induit certes une augmentation substantielle du taux de stockage mais qui évolue lentement (i.e. évolution quadratique). L'algorithme DC-H ne nécessite aucun stockage mémoire, ni aucun temps de pré-calcul étant donné qu'il ne matérialise aucun résultat partiel. Cependant, $HSky$ surpasse DC-H en terme de temps de réponse aux requêtes (Figure 4c). En effet, $HSky$ affiche des temps de réponse quasi-instantanés alors que pour DC-H ils augmentent sensiblement avec le nombre de dimensions hiérarchiques. Cela s'explique par le fait que DC-H recalcule à chaque nouvelle requête l'ensemble des skyline, alors qu'il est déduit à partir de simples opérations de soustraction et d'union ensembliste dans $HSky$. Le temps de réponse aux requêtes est un critère d'évaluation très important car nous nous positionnons dans un contexte d'analyse en ligne.

Variation de la taille de l'ensemble de données Dans cette expérimentation, le nombre de tuples de l'ensemble de données varie de 50.000 à 700.000. La Figure 5 montre que la taille mémoire et le temps de pré-calcul de $HSky$ augmentent avec la taille de l'ensemble de données. Cela s'explique par le fait que le volume des informations stockées et analysées croît avec l'augmentation du volume des données (i.e. les points skyline stockés dans les ensembles

$HNSky$), ce qui induit une évolution polynomiale (i.e. lente) du taux de stockage. Cependant, notre méthode est plus performante que DC-H en terme de temps de réponse aux requêtes.

Variation du nombre de niveaux hiérarchiques des dimensions hiérarchiques Nous avons fait varier le nombre de niveaux hiérarchiques des dimensions hiérarchiques de 3 à 7. La Figure 6 montre que la taille mémoire et le temps de pré-calcul de $HSky$ augmentent avec le nombre de niveaux hiérarchiques. Cette augmentation peut s'expliquer par l'évolution exponentielle du nombre de valeurs des dimensions hiérarchiques induite par l'augmentation du nombre de niveaux hiérarchiques associé à ces dernières. Nous pouvons cependant noter que ces expérimentations ont été poussées assez loin étant donné que dans les applications réelles il est rare de trouver des dimensions avec 7 niveaux hiérarchiques. Une fois de plus, notre méthode est plus performante que DC-H en terme de temps de réponse aux requêtes.

La construction de la structure de $HSky$ induit un coût non négligeable en terme de temps de pré-calcul et de stockage mémoire. Pour que l'utilisation de cette structure soit bénéficiaire, il faut un certain nombre d'interactions utilisateur (6 à 8 requêtes en moyenne).

5 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode $HSky$ qui étend la recherche des points skyline aux dimensions hiérarchiques. La définition de hiérarchies sur les dimensions permet à l'utilisateur d'affiner ou de généraliser ses préférences lors du calcul des skyline. Nous avons mis en évidence un ensemble de propriétés permettant la formalisation des relations hiérarchiques entre les préférences associées aux différentes dimensions. Ces propriétés permettent notamment aux utilisateurs de naviguer le long des relations hiérarchiques entre les préférences tout en assurant un calcul en ligne des points skyline correspondants. En particulier, la propriété de monotonie hiérarchique facilite l'analyse et le calcul des points skyline en présence de préférences hiérarchiques. Cette propriété permet de caractériser les points skyline qui restent skyline et ceux qui ne le sont plus, lors de la spécialisation (drill-down) ou de la généralisation (roll-up) des préférences hiérarchiques. Le nombre de tests de dominance est ainsi considérablement réduit. De plus, la structure de données $HSky$ proposée permet un calcul efficace des skyline lors de la navigation sur les dimensions hiérarchiques. Les expérimentations réalisées soulignent l'efficacité des performances de $HSky$ comparé à la méthode DC-H dès que le nombre de requêtes d'une navigation est supérieure à un certain seuil (6 à 8 requêtes en moyenne). Cependant, cette structure a vocation à évoluer selon le contexte d'application. Par exemple, si la taille de la structure est trop importante, il est possible de restreindre les spécialisations / généralisations aux valeurs explicitement ordonnées dans la préférence de base. Nous souhaitons étudier le cas des dimensions hiérarchiques associées à des préférences dynamiques comme traité dans Bouadi et al. (2013, 2012). Dans ce cas de figure l'aspect dynamique des préférences peut occasionner une explosion du nombre de préférences à matérialiser. Une piste que nous souhaitons explorer est de restreindre les préférences spécialisables / généralisables à une classe de préférences particulières qui se sont avérés intéressantes pour le raffinement de préférences non hiérarchiques Bouadi et al. (2013).

Références

Antony, S., P. Wu, D. Agrawal, et A. E. Abbadi (2009). Aggregate skyline : Analysis for online users. In *Proceedings of the 2009 Ninth Annual International Symposium on Applications and the Internet*, pp. 50–56. IEEE Computer Society.

Antony, S., P. Wu, D. Agrawal, et A. El Abbadi (2008). Moolap : Towards multi-objective olap. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pp. 1394–1396. IEEE Computer Society.

Borzsonyi, S., D. Kossmann, et K. Stocker (2001). The skyline operator. In *Proc of the 17th International Conference on Data Engineering*, pp. 421–430. IEEE Computer Society.

Bouadi, T., M. Cordier, et R. Quiniou (2012). Incremental computation of skyline queries with dynamic preferences. In *DEXA (1)*, pp. 219–233.

Bouadi, T., M.-O. Cordier, et R. Quiniou (2013). Computing skyline incrementally in response to online preference modification. *T. Large-Scale Data- and Knowledge-Centered Systems 10*, 34–59.

Huang, Z., J. Guo, S. Sun, et W. Wang (2008). Efficient optimization of multiple subspace skyline queries. *J. Comput. Sci. Technol.*, 103–111.

Jin, W., M. Ester, Z. Hu, et J. Han (2007a). The multi-relational skyline operator. In *ICDE*, pp. 1276–1280.

Jin, W., A. K. H. Tung, M. Ester, et J. Han (2007b). On efficient processing of subspace skyline queries on high dimensional data. In *Proc of the 19th International Conference on Scientific and Statistical Database Management*, pp. 12. IEEE Computer Society.

Magnani, M. et I. Assent (2013). From stars to galaxies : skyline queries on aggregate data. In *EDBT*, pp. 477–488.

Raïssi, C., J. Pei, et T. Kister (2010). Computing closed skycubes. *Proc. VLDB Endow.*, 838–847.

Tao, Y., X. Xiao, et J. Pei (2008). Efficient skyline and top-k retrieval in subspaces. *IEEE Trans. on Knowl. and Data Eng.*, 1072–1088.

Trenkler, G. (1994). Univariate discrete distributions : N.l. johnson, s. kotz and a.w. kemp (1992) : 2nd edition. new york : John wiley, isbn 0-471-54897-9. *Computational Statistics & Data Analysis*, 240–241.

Wong, R. C. W., J. Pei, A. W. C. Fu, et K. Wang (2009). Online skyline analysis with dynamic preferences on nominal attributes. *IEEE Trans. on Knowl. and Data Eng.*, 35–49.

Summary

Skyline queries represent a powerful tool for multidimensional data analysis and for decision aid. This article proposes a method that extends skyline queries to multiple hierarchical dimensions.